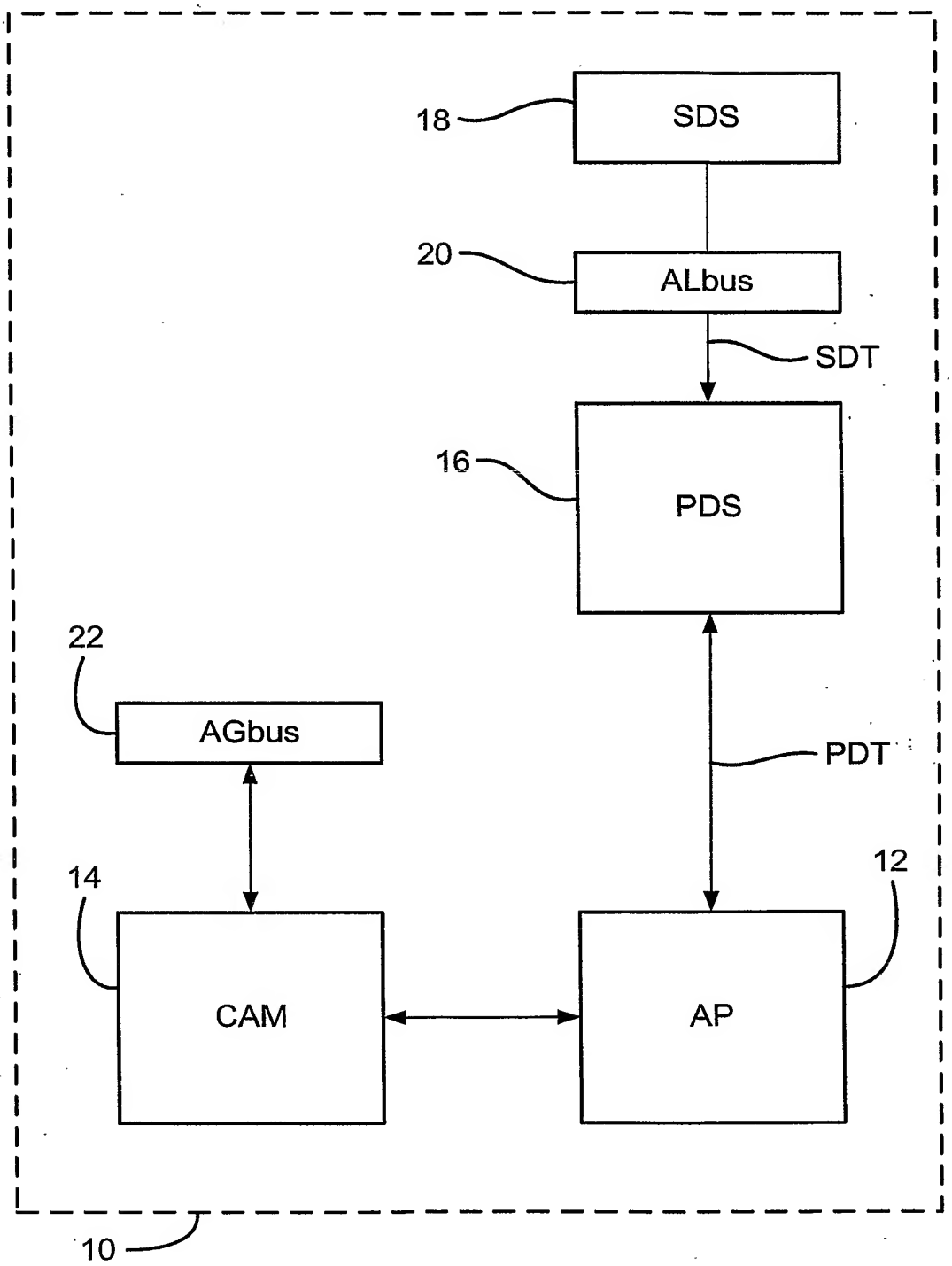


Fig.1

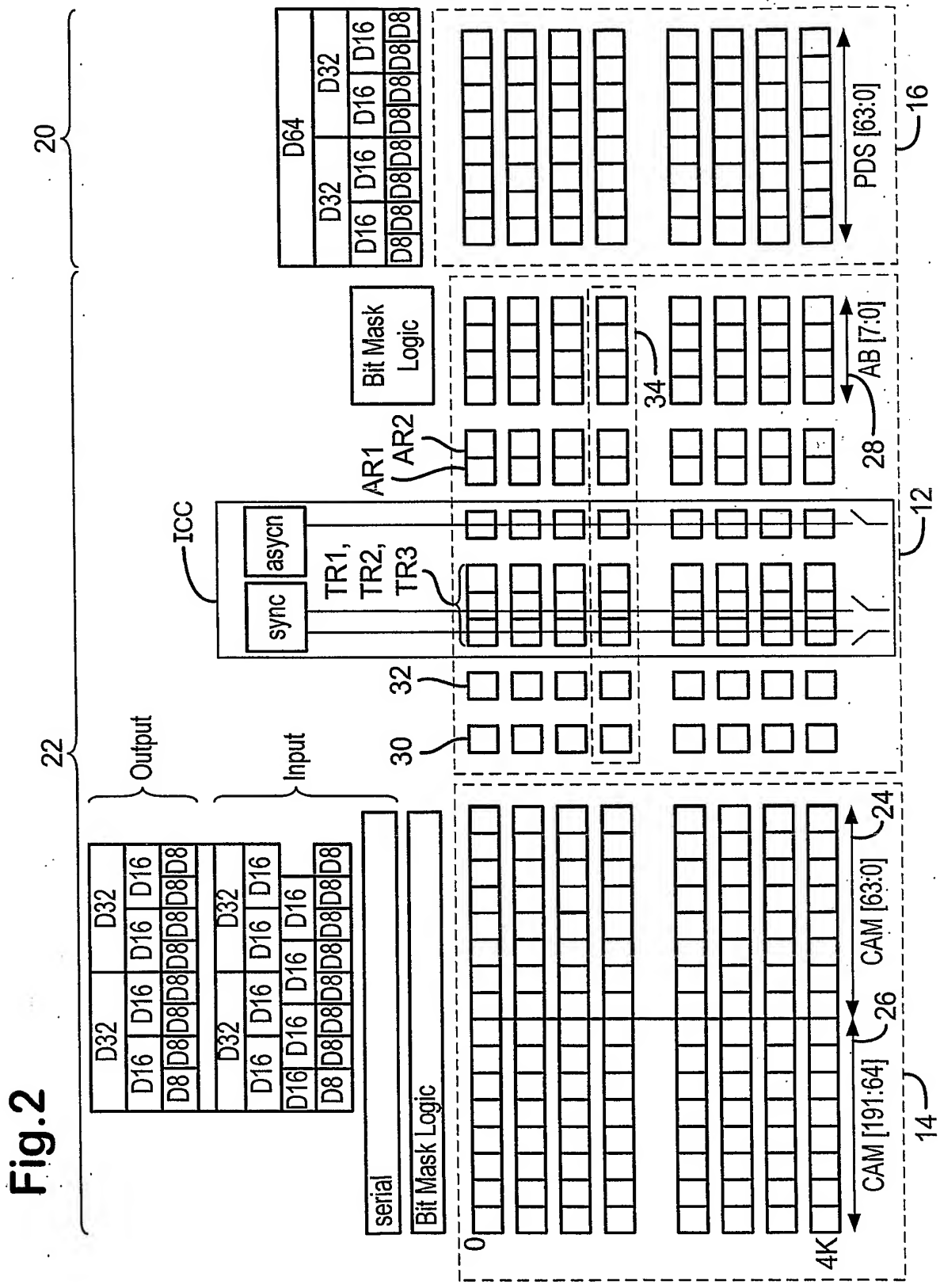

20
22
D64
D32
D16
D8
PDS [63:0]
16
Bit Mask Logic
AB [7:0]
34
28
AR1
AR2
ICC
sync
asycn
TR1, TR2, TR3
12
32
30
Output
Input
serial
Bit Mask Logic
0
4K
CAM [63:0]
24
26
CAM [191:64]
14

Fig.2

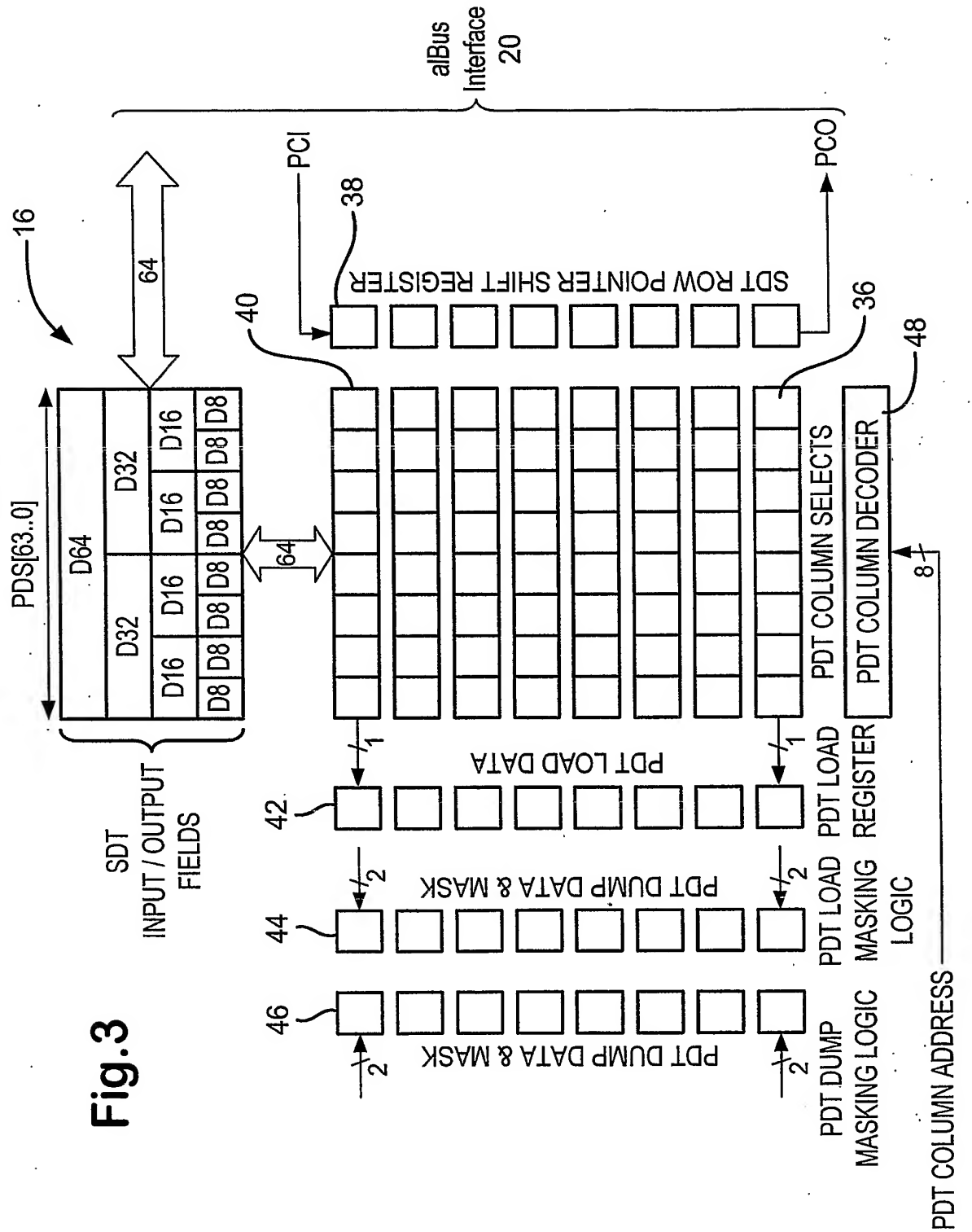

aiBus Interface 20
PCI
PCO
38
16
64
SDT ROW POINTER SHIFT REGISTER
40
36
48
PDS[63..0]
D64
D32
D16
D8
64
PDT COLUMN SELECTS
PDT COLUMN DECODER
8
PDT LOAD DATA
1
SDT INPUT / OUTPUT FIELDS
42
PDT LOAD REGISTER
PDT DUMP DATA & MASK
2
44
PDT LOAD MASKING LOGIC
46
PDT DUMP MASKING LOGIC
PDT COLUMN ADDRESS

Fig.3

Fig.4

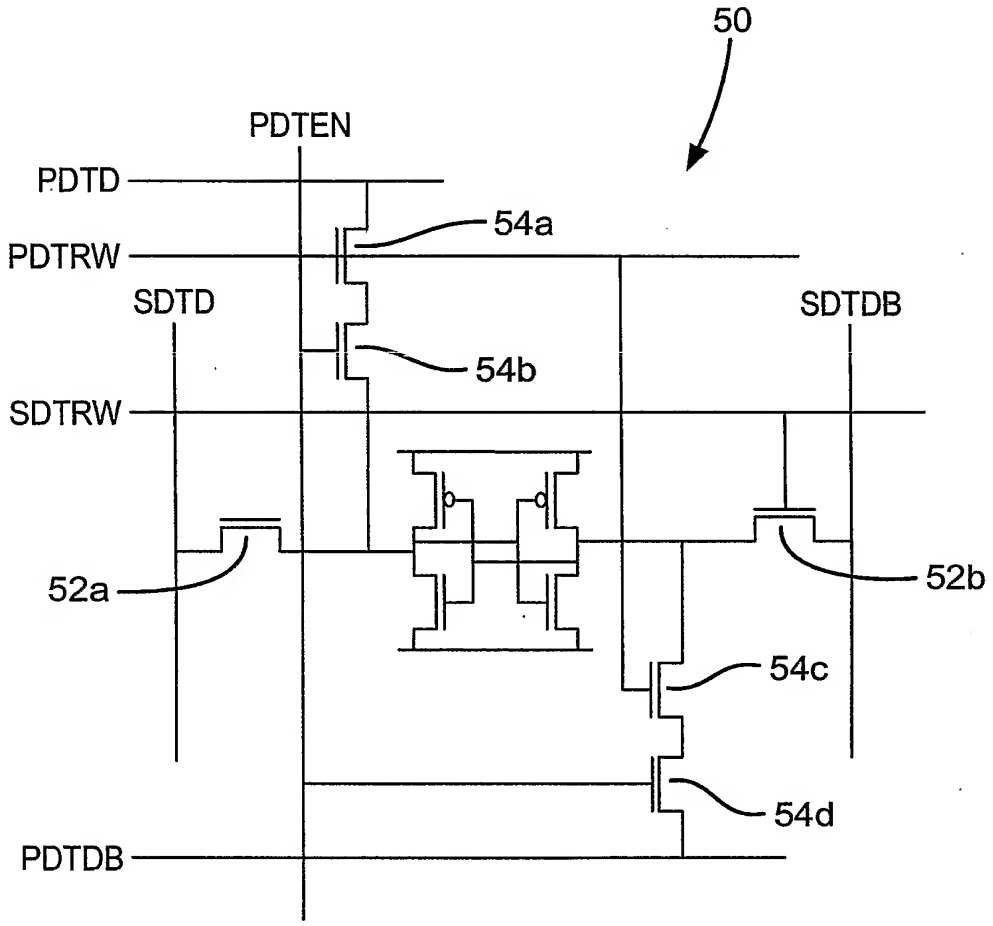
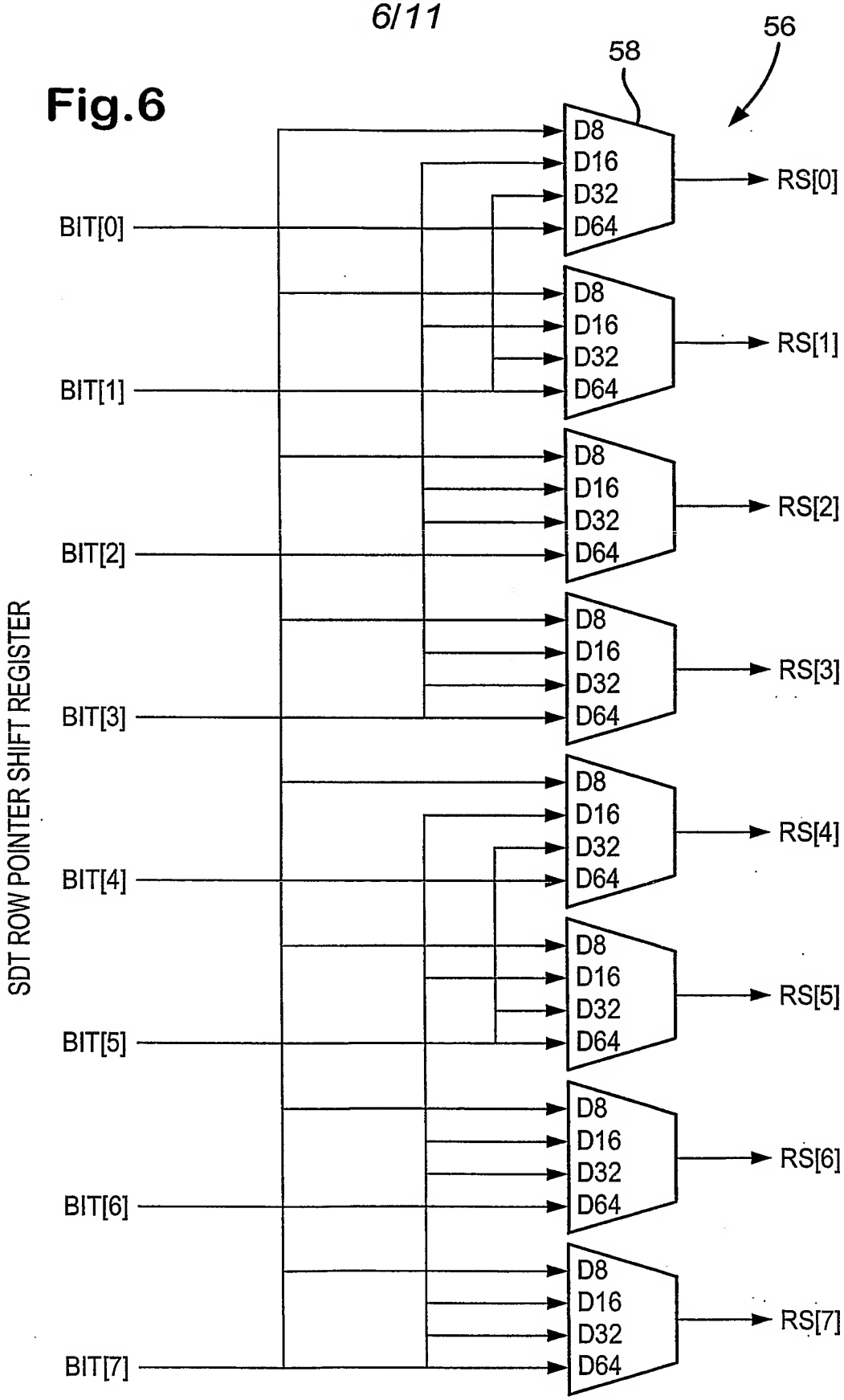

50
PDTEN
PDTD
54a
PDTRW
SDTD
SDTDB
54b
SDTRW
52a
52b
54c
54d
PDTDB

Fig.5

row address MOD 8	Row Strobe logic conditions
0	SDTRW_A[0] = RS[0]
	SDTRW_B[0] = ~D8.RS[0]
	SDTRW_C[0] = (D32+D64).RS[0]
	SDTRW_D[0] = D64.RS[0]
1	SDTRW_A[1] = (D8+D64).RS[1]
	SDTRW_B[1] = (D16+D64).RS[1]
	SDTRW_C[1] = (D32+D64).RS[1]
	SDTRW_D[1] = D64.RS[1]
2	SDTRW_A[2] = ~D16.RS[2]
	SDTRW_B[2] = (D16+D64).RS[2]
	SDTRW_C[2] = (D32+D64).RS[2]
	SDTRW_D[2] = D64.RS[2]
3	SDTRW_A[3] = (D8+D64).RS[3]
	SDTRW_B[3] = ~D8.RS[3]
	SDTRW_C[3] = (D32+D64).RS[3]
	SDTRW_D[3] = D64.RS[3]
4	SDTRW_A[4] = (D8+D64).RS[4]
	SDTRW_B[4] = ~D8.RS[4]
	SDTRW_C[4] = (D32+D64).RS[4]
	SDTRW_D[4] = D64.RS[4]
5	SDTRW_A[5] = ~D16.RS[5]
	SDTRW_B[5] = (D16+D64).RS[5]
	SDTRW_C[5] = (D32+D64).RS[5]
	SDTRW_D[5] = D64.RS[5]
6	SDTRW_A[6] = (D8+D64).RS[6]
	SDTRW_B[6] = (D16+D64).RS[6]
	SDTRW_C[6] = (D32+D64).RS[6]
	SDTRW_D[6] = D64.RS[6]
7	SDTRW_A[7] = RS[7]
	SDTRW_B[7] = ~D8.RS[7]
	SDTRW_C[7] = (D32+D64).RS[7]
	SDTRW_D[7] = D64.RS[7]

Table 2

Fig.6


56
58
SDT ROW POINTER SHIFT REGISTER
BIT[0]
BIT[1]
BIT[2]
BIT[3]
BIT[4]
BIT[5]
BIT[6]
BIT[7]
D8
D16
D32
D64
RS[0]
RS[1]
RS[2]
RS[3]
RS[4]
RS[5]
RS[6]
RS[7]

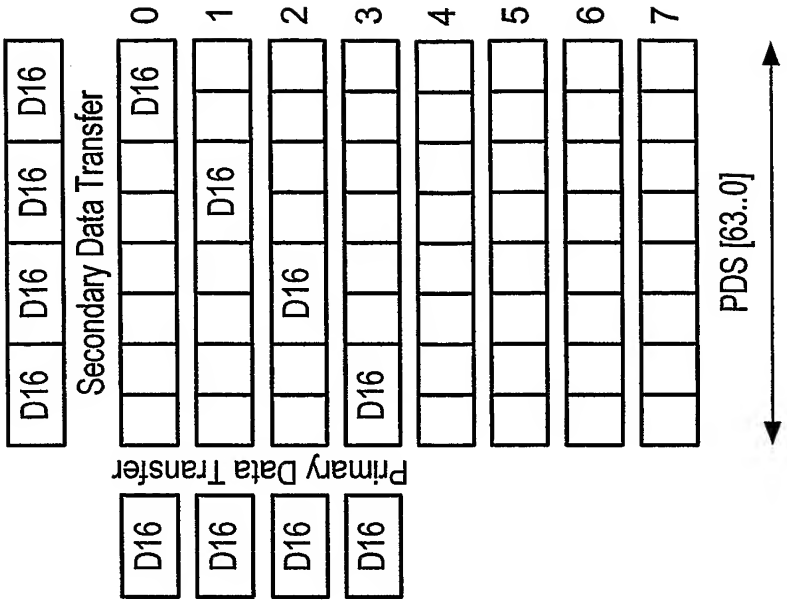

0
1
2
3
4
5
6
7
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
D16
Secondary Data Transfer
Primary Data Transfer
PDS [63..0]

Fig.7 (b)

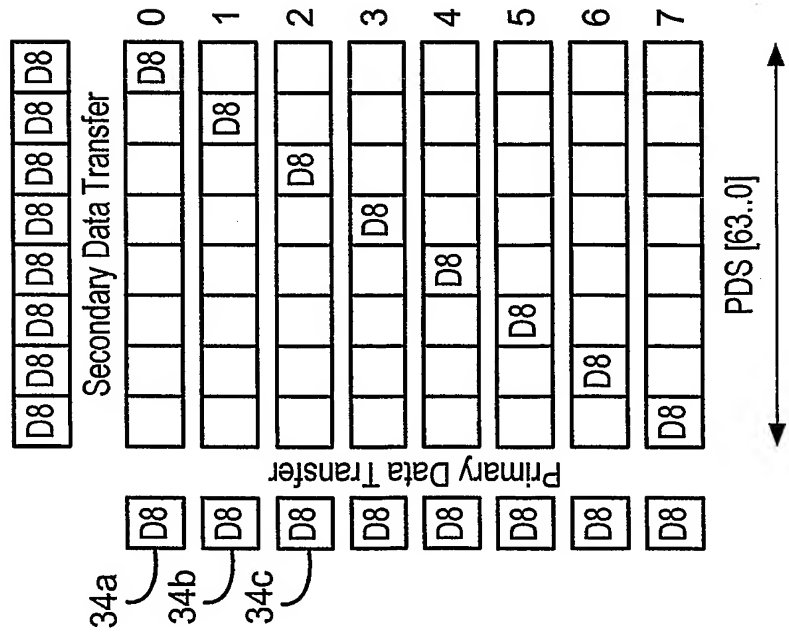

0
1
2
3
4
5
6
7
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
D8
Secondary Data Transfer
Primary Data Transfer
PDS [63..0]
34a
34b
34c

Fig.7 (a)

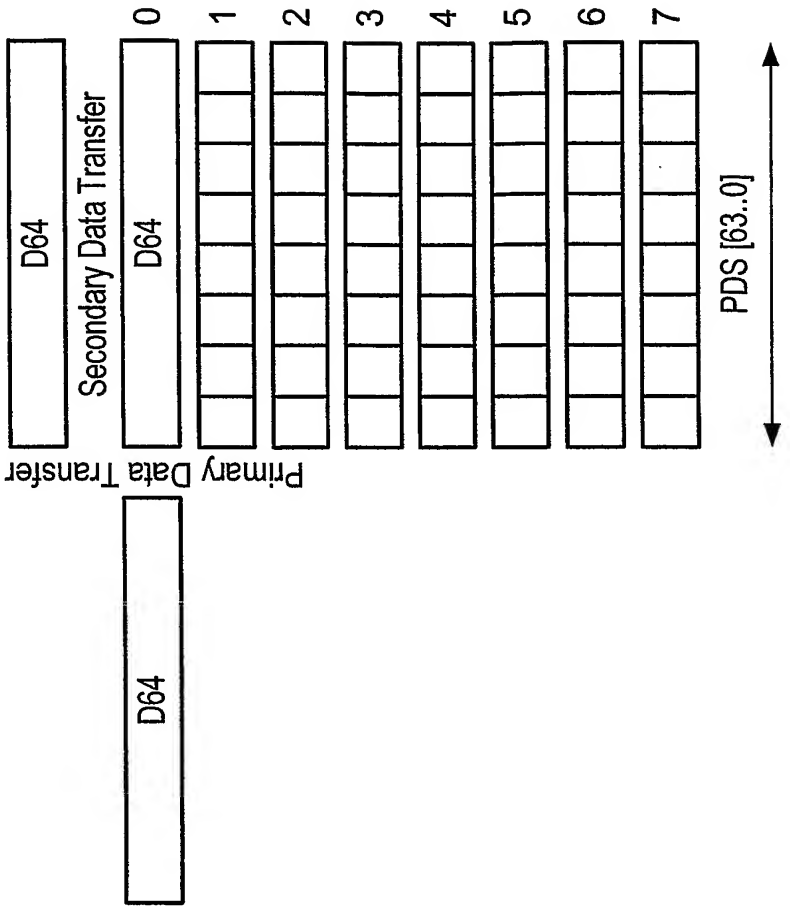

0
1
2
3
4
5
6
7
D64
Secondary Data Transfer
D64
PDS [63..0]
Primary Data Transfer
D64

Fig.7 (d)

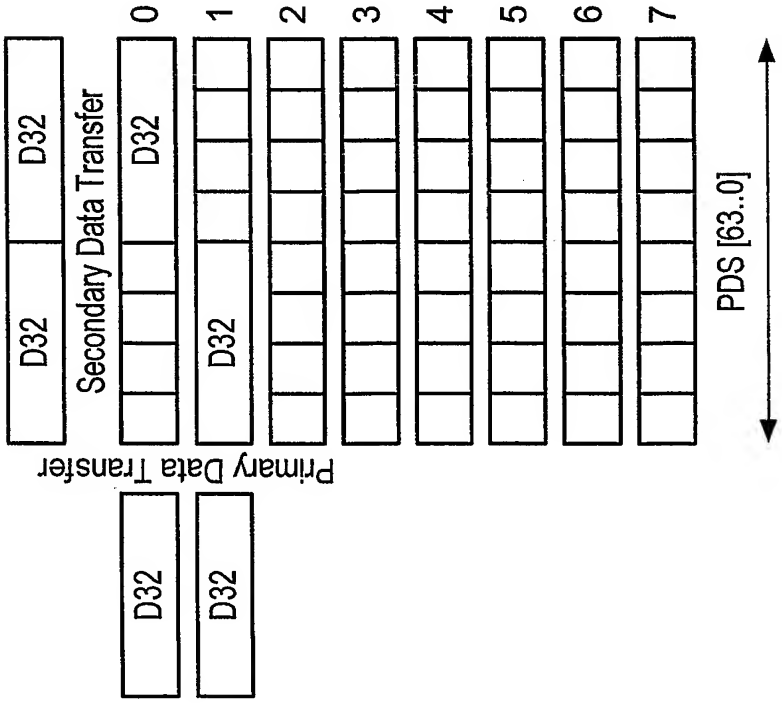

0
1
2
3
4
5
6
7
D32
D32
Secondary Data Transfer
D32
D32
PDS [63..0]
Primary Data Transfer
D32
D32

Fig.7 (c)

Fig.8

byte address	column address	Row Strobe logic conditions
0	7..0	PDTEN_A[7..0] = BS[0].CS[7..0]
		PDTEN_B[7..0] = ~D8.BS[0].CS[7..0]
		PDTEN_C[7..0] = (D32+D64).BS[0].CS[7..0]
		PDTEN_D[7..0] = D64.BS[0].CS[7..0]
1	15..8	PDTEN_A[15..8] = (D8+D64).BS[1].CS[15..8]
		PDTEN_B[15..8] = ~D8.BS[1].CS[15..8]
		PDTEN_C[15..8] = (D32+D64).BS[1].CS[15..8]
		PDTEN_D[15..8] = D64.BS[1].CS[15..8]
2	24..16	PDTEN_A[24..16] = ~D16.BS[2].CS[24..16]
		PDTEN_B[24..16] = (D16+D64).BS[2].CS[24..16]
		PDTEN_C[24..16] = (D32+D64).BS[2].CS[24..16]
		PDTEN_D[24..16] = D64.BS[2].CS[24..16]
3	31..25	PDTEN_A[31..25] = (D8+D64).BS[3].CS[31..25]
		PDTEN_B[31..25] = ~D8.BS[3].CS[31..25]
		PDTEN_C[31..25] = (D32+D64).BS[3].CS[31..25]
		PDTEN_D[31..25] = D64.BS[3].CS[31..25]
4	39..32	PDTEN_A[31..25] = (D8+D64).BS[4].CS[39..32]
		PDTEN_B[31..25] = ~D8.BS[4].CS[39.32]
		PDTEN_C[31..25] = (D32+D64).BS[4].CS[39..32]
		PDTEN_D[31..25] = D64.BS[4].CS[39..32]
5	47..40	PDTEN_A[24..16] = ~D16.BS[5].CS[47..40]
		PDTEN_B[24..16] = (D16+D64).BS[5].CS[47..40]
		PDTEN_C[24..16] = (D32+D64).BS[5].CS[47..40]
		PDTEN_D[24..16] = D64.BS[5].CS[47..40]
6	55..48	PDTEN_A[15..8] = (D8+D64).BS[6].CS[55..48]
		PDTEN_B[15..8] = ~D8.BS[6].CS[55..48]
		PDTEN_C[15..8] = (D32+D64).BS[6].CS[55..48]
		PDTEN_D[15..8] = D64.BS[6].CS[55..48]
7	63..56	PDTEN_A[7..0] = BS[7].CS[63..56]
		PDTEN_B[7..0] = ~D8.BS[7].CS[63..56]
		PDTEN_C[7..0] = (D32+D64).BS[7].CS[63..56]
		PDTEN_D[7..0] = D64.BS[7].CS[63..56]

Table 4

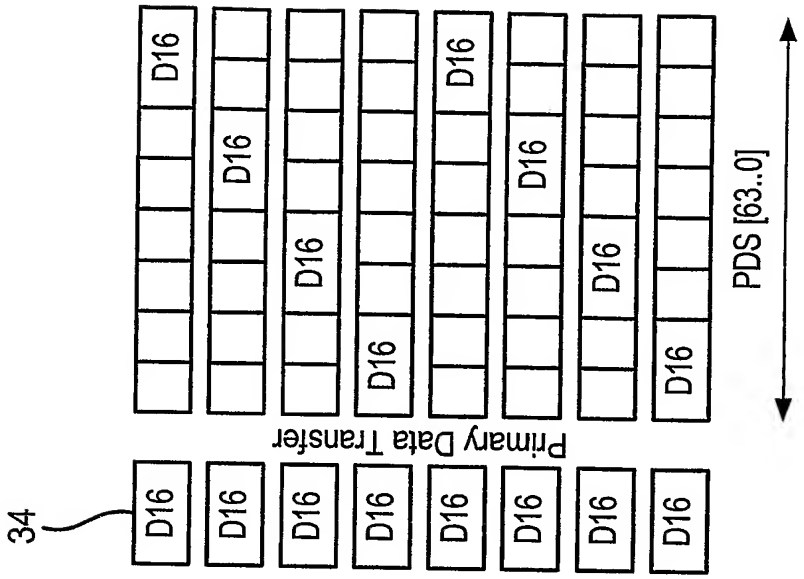

D16
PDS [63..0]
Primary Data Transfer
34

Fig.9 (b)

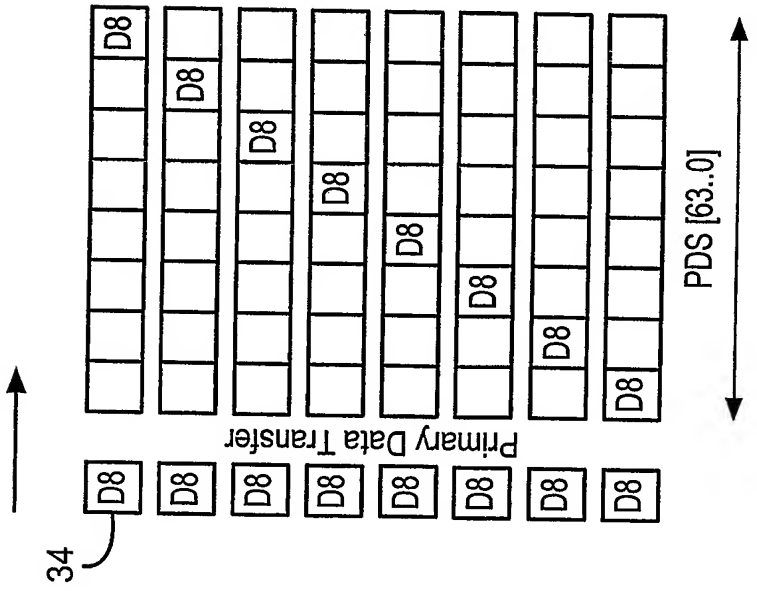

D8
PDS [63..0]
Primary Data Transfer
34

Fig.9 (a)

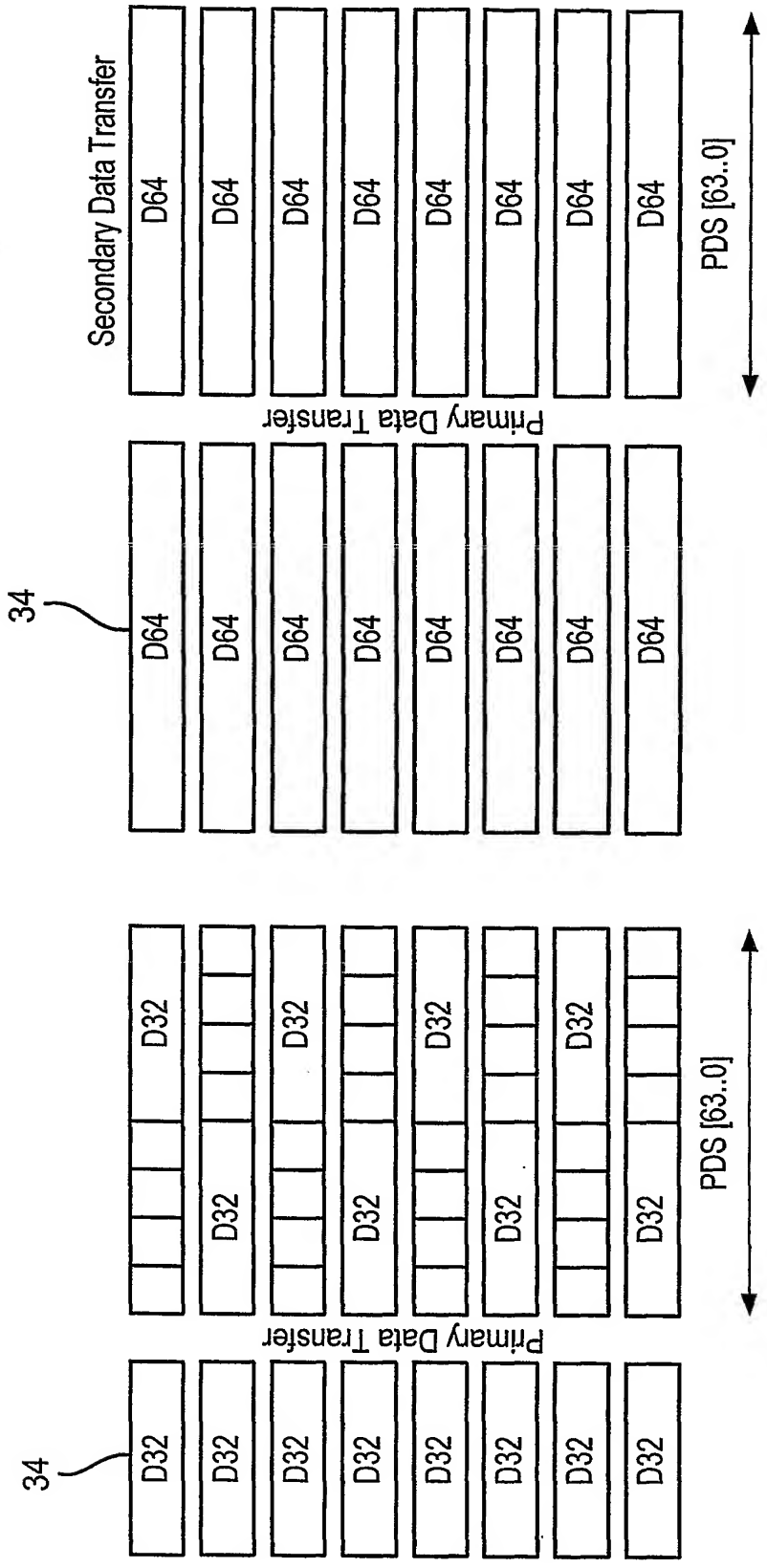

Secondary Data Transfer
D64
PDS [63..0]
Primary Data Transfer
34
D32

Fig.9 (c)

Fig.9 (d)